

quick and handy REFERENCE pamphlet

FOR
EVERGREEN'S TIMESHARING COMPUTER SYSTEM

CONTENTS

- * ASCII Code Chart
- * Statement Types for BASIC75
- * Special Characters
- * Return Variables Legend
- * Operators for BASIC75
- * System Commands
- * Protect Level Chart
- * Functions

ASCII CODE CHART

BITS				CONTROL				HIGH X & Y GRAPHIC INPUT				LOW X GRAPHIC INPUT				LOW Y GRAPHIC INPUT					
B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀														
0	0	0	0	0	0	0	0	NUL	0	DLE	16	SP	32	0	48	@	P	96	p	112	
0	0	0	0	0	0	1	0	SOH	1	DC1	17	!	33	1	49	A	Q	97	a	q	113
0	0	0	0	1	0	0	0	STX	2	DC2	18	"	34	2	50	B	R	98	b	r	114
0	0	0	0	1	1	0	0	ETX	3	DC3	19	#	35	3	51	C	S	99	c	s	115
0	0	1	0	0	0	0	0	EOT	4	DC4	20	\$	36	4	52	D	T	100	d	t	116
0	0	1	0	0	1	0	0	ENQ	5	NAK	21	%	37	5	53	E	U	101	e	u	117
0	0	1	0	1	0	0	0	ACK	6	SYN	22	&	38	6	54	F	V	102	f	v	118
0	0	1	0	1	1	0	0	BEL	7	ETB	23	'	39	7	55	G	W	103	g	w	119
0	1	0	0	0	0	0	0	BS	8	CAN	24	(	40	8	56	H	X	104	h	x	120
0	1	0	0	0	1	0	0	BACK SPACE													
0	1	0	0	1	0	0	0	HT	9	EM	25	)	41	9	57	I	Y	105	i	y	121
0	1	0	1	0	0	0	0	LF	10	SUB	26	*	42	:	58	J	Z	106	j	z	122
0	1	0	1	0	1	0	0	LINE FEED													
0	1	0	1	1	0	0	0	VT	11	ESC	27	+	43	;	59	K	[	107	k	{	123
0	1	1	0	0	0	0	0	FF	12	FS	28	,	44	<	60	L	\	108	l	!	124
0	1	1	0	1	0	0	0	CR	13	GS	29	-	45	=	61	M	]	109	m	}	125
0	1	1	0	1	1	0	0	RETURN													
0	1	1	1	0	0	0	0	SO	14	RS	30	.	46	>	62	N	^	110	n	~	126
0	1	1	1	1	0	0	0	SI	15	US	31	/	47	?	63	O	_	111	o		127
1	1	1	1	1	1	1	1													RUBOUT (DEL)	

A COMPLETE LIST OF STATEMENTS

FOR THE BASIC75 PROGRAMMING LANGUAGE

STATEMENTS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
ASSIGN	128 ASSIGN "FILE",1,Z9 162 ASSIGN A\$,A,Q,M\$ 170 ASSIGN A\$,-1,X 1620 ASSIGN *,2	To change the file referred to by a specified file number. Returns 0 if successful. Several variants are recognized: M\$ is a mask encoding string. A negated file number indicates the file is READ-ONLY status. Deassigns the specified file number.
CHAIN	300 CHAIN "PROG" 310 CHAIN "PROG.E111" 320 CHAIN "\$PROG",80 330 CHAIN A\$,A 340 CHAIN "PROG", Q+14	Retrieves program specified from proper library, and begins executing at beginning, or line number if specified (line number ignored if program specified is on another ID and release locked). The current program is destroyed, except for COM mon variables.
COMmon	360 COM A,B1,C(20),C\$(120)	Declares variables to be in common; they can then be passed via CHAIN. Variables must agree in size and type with program chained to. Must be lowest numbered statement.
CREATE	362 CREATE T,"FLYER",5,128 363 CREATE X,A\$,R	Opens a file (FLYER,A\$) to a specified length (5,R) with an optional record size (128). The return variable (T,X) is set to 0 if successful. The protect status of a CREATED file is RL.
DATA	370 DATA 99,10.7,"HIYA!"	Specifies DATA; Read from left to right.
DEF	10 DEF FND(A)= COS(A)*3	Defines a function of one numerical argument.
DIM	380 DIM A(72),A\$(255)	Specifies maximum array or string size.
DO	390 IF A THEN DO 400 ELSE DO 405 UNTIL X = 0 DO	Specifies the beginning of a block of statements. Can only follow THEN, ELSE, WHILE or UNTIL.
DOEND	410 DOEND	Specifies the end of a block of statements started by a DO.
ELSE	420 ELSE PRINT A 430 ELSE DO	Specifies a statement to be executed if the evaluation of an IF...THEN was false. Must immediately follow an IF...THEN or a block specified by IF-THEN-DO-DOEND. May start a block.

STATEMENTS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
END	440 END	Terminates the program. Last statement of a program must be END.
ENTER	450 ENTER #P 460 ENTER A,B,C 470 ENTER #P,A,B,C 475 ENTER 25,B,C\$	Fills the first variable (#P) with the user port number and/or allows the user a specified number of seconds to reply (A,25), returns the actual response time (B), and returns the value entered (C,C\$). On time-out the response time is set to -256. On illegal input type, the response time is negated. Leading blanks, quotes, and null entries are accepted as valid string characters by ENTER.
FOR...NEXT	480 FOR J=1 TO N STEP 3 490 FOR J=1,2,4,4,12,B	Executes the statements between FOR and NEXT the specified number of times (a loop) incrementing the variable (J) by a step number (3) (or by 1 if step is not given) until the FOR variable (J) is greater than the limit value (N). See NEXT. Executes the statements between FOR and NEXT, setting the FOR variable (J) to the NEXT value in the list (1,2...,B) until the list is exhausted.
GOTO	500 GOTO 900	Transfers control (jumps) to the specified line number.
GOTO...OF	510 GOTO N OF 100,10,700	Transfers control to the Nth statement of the statements listed after OF.
GOSUB	470 GOSUB 800	Begins executing the subroutine at the specified statement (see STACK and RETURN)
GOSUB...OF	480 GOSUB N OF 100,10,700	Begins executing the Nth subroutine of the subroutines listed after OF.
IF...THEN	490 IF A#10 THEN 350 500 IF B >=5 THEN PRINT B 510 IF C < D THEN DO	Logical test. Transfers control to statement number (350). If true, or executes statement following THEN (PRINT B) or begins execution of DO block (DO). If DO block specified and evaluation is false, control is transferred to statement following balancing DOEND.
IMAGE	520 IMAGE #,6D,AA	Used to specify the format of a PRINT USING statement.
INPUT	530 INPUT X\$,A2,B4	Allows data to be entered from terminal while a program is running.

STATEMENTS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
LET	540 LET A=B=0 550 A=B*5 552 A=C+7,Z=COS(A1)	Assigns values to variables. LET is optional. Multiple assignments on a line are separated by commas
LINPUT	560 LINPUT A 570 LINPUT C\$	Similar to ENTER. No prompt (?) printed no timing.
LOCK	572 LOCK #1 573 LOCK #A,T	Allows programmatic lock-out of files. Only one user may have a file locked. If no return variable is specified, an attempt to LOCK a file currently LOCKED by another user will cause the latter program to suspend until the former program UNLOCKS the file. If a return variable is used (T), a LOCK attempt will not suspend, and a successful LOCK will return 0.
NEXT	580 NEXT J	Marks the end of a FOR loop. See FOR.
PAGE	590 PAGE A SAMPLE HEADING	On paginated listing, causes a skip to a new page and text after PAGE to be printed in heading. Ignored during execution.
PERFORM	107 PERFORM SYS(13) 165 PERFORM BRK(0),PUSH(X) 167 PERFORM A=7,B=C1,ERR(1)	Evaluates expressions, separated by commas, for their side effects rather than for the values they return. Also evaluates and executes assignment statements.
PCALL	600 PCALL 505	Procedure call. Performs a gosub to the specified line number (505) pushing the current value of the Q register onto the user-defined data stack and setting the value of the Q register to the new top of the data stack.
PRINT	25 PRINT "Y=";Y;"VOLTS"	Evaluates and prints variables, expressions, and text on the user's terminal.
	30 PRINT #F,R; A\$,B,Z	Prints on a specified file and record number.
	40 PRINT #F,R	Prints an End-Of-Record marker on file number F, record R.
PRINT USING	105 PRINT USING 30;A,B	Allows formatted output to the user's terminal with the format specifier either explicit or in an IMAGE statement

STATEMENTS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
PURGE	600 PURGE T,"FILE2" 605 PURGE X,A\$	Deletes the specified entry (FILE22,A\$) From the users catalog. Return variable (T,X) is set to 0 for a successful PURGE.
READ	110 READ A,B,C 115 READ #1; A7,C\$	Assigns values to variables. Values come from either DATA statements or a file.
REMARK	112 REMARK: A COMMENT 256 REM READ VOLTAGE	Allows the use of comments and remarks in a program. Can be abbreviated REM.
RESTORE	310 RESTORE 320 RESTORE 410	Restores the DATA pointer to the first DATA statement or a specified statement.
RESTORE...OF	350 RESTORE N OF 1000,340,370	Restores the DATA pointer to the Nth line specified.
RETURN	410 RETURN 510 RETURN Z 520 RETURN Q-4*T	Returns from a subroutine called by a GOSUB statement. Returns from a procedure (possibly recursive) called by a PCALL statement. The expression refers to how many data stack elements are to be popped.
SPAWN	700 SPAWN T,"PROG" 710 SPAWN T,A\$	Starts the named program executing on a background port. Return variable (T) is set to the background port number assigned, or 0 if none available (see the JOB command).
STACK	720 STACK *,20 730 STACK A,5 740 STACK A,B	Specifies the maximum size of the GOSUB stack (Default is 9). Only one STACK * statement allowed in a program. Assigns the specified array (A) as the current stack, and sets the initial stack pointer (5,B). The array must be DIMensioned.
STOP	750 STOP	Terminates the program, may be used anywhere in the program.
SUSPEND	760 SUSPEND 30 770 SUSPEND A*B	Causes the currently executing program to be suspended for the specified number of seconds (30,A*B).
UNLOCK	772 UNLOCK #1 776 UNLOCK #A,T	Unlocks a previously LOCKED file. Posts the current record. Return variable (T) is set to 0 for a successful UNLOCK.

STATEMENTS

NAME	EXAMPLE	PURPOSE
UNTIL...DOEND	780 UNTIL TYP(1)=3 DO 790 UNTIL A=0 DO	Causes the statements between UNTIL and the balancing DOEND to be executed (a loop) until the specified condition (TYP(1)=3,A=0) is true. The first time UNTIL is executed the expression is not evaluated, so execution of the loop will always occur at least once.
USING	800 USING 400	Executes the statement specified (400) instead of the USING statement.
USING...OF	810 USING N OF 100,10,700	Executes the Nth statement of the list instead of the USING statement.
WHILE...DOEND	820 WHILE A=0 DO 830 WHILE B<5 DO	Similar to UNTIL, except the loop will be executed while the expression (A=0,B<5) is true. Will not necessarily execute the loop at all. At 820 if A#0 control will transfer past the balancing DOEND.

SPECIAL CHARACTERS

NOTE: Superscript "c" indicates a control character. Hold the control key (usually marked as CTRL) down while striking appropriate character.

KEY	FUNCTION
Break	Interrupts a program or listing if pressed. Terminates program when entered in response to an input request (an INPUT, ENTER, or LINPUT statement).
C ^c	Terminates program when entered in response to an input request statement.
X ^c	Deletes the current line being typed.
R ^c	Repeats the current line being typed. The line is edited to remove characters deleted by backspace (←).
U ^c	Suspends the output of a running program until another key is pressed.
D ^c	Suspends all output from a program until an input request statement is executed.
Carriage-Return	Marked as CR or RETURN. Must follow every command, statement, or data entry. Causes terminal typeface to return to the first print position. Computer responds with a linefeed and a prompt ("> ").
←	Backspace. Seen as Shift O on Teletypes, and as an underline on other terminals. Deletes one character for each typed in the current line.

RETURN VARIABLE VALUES
FOR BASIC75 STATEMENT TYPES

ASSIGN

- 0 - Successful assignment
- 1 - File is READ-ONLY
- 2 - File is READ-ONLY and PROTECTED
- 3 - Non-existent file requested
- 4 - Insufficient file buffers reserved in a FILES statement
- 5 - Record size larger than previous file
- 6 - Entry is a program, not a file

CREATE

- 0 - Successful create
- 1 - Duplicate entry
- 2 - Invalid name
- 3 - Library space full
- 4 - System overload
- 5 - Invalid record size
- 6 - Invalid number of words per record
- 7 - File killed while being created
- 8 - Unsuccessful. Purge and re-create

PURGE

- 0 - Successful purge
- 1 - File in use
- 2 - Unsuccessful. attempt to re-purge
- 3 - Non-existent file or program requested

LOCK

- 0 - Successful
- 1 - File previously locked
- 2 - Unsuccessful

UNLOCK

- 0 - Successful
- 1 - Not previously locked

A COMPLETE LIST OF OPERATORS IN BASIC75

Arithmetic Operators

<u>SYMBOL</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
=	100 A=15,Z\$="ALP" 200 B=6,A=J4=1.27	Assignment operator. Assigns a value to a variable. Multiple assignments on a line are separated by commas or semicolons.
+	127 A=B+C+D	Addition
-	128 F=G1-A(1,2)	Subtraction
*	129 J=C(I,J)*4	Multiplication
/	130 K=K/2	Division
↑	131 F=X↑2	Exponentiation. As in X^2
MOD	132 G=A MOD B	Modulus. Remainder of A/B.
MIN	137 E=F MIN J	Smaller of F and J
MAX	254 J=K MAX Z	Larger of K and Z

Relational Operators

Note: The numeric values used in logical evaluation are: "true" is any nonzero number, "false" is zero (0).

<u>SYMBOL</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
=	122 If A=B THEN 305	<u>expression</u> "equals" <u>expression</u>
#	256 If D#(2*D4) THEN STOP	<u>exp</u> "doesn't equal" <u>exp</u>
<>	512 If C <> D THEN READ C	<u>exp</u> "doesn't equal" <u>exp</u>
>	(Examples are similar)	<u>exp</u> "is greater than" <u>exp</u>
<		<u>exp</u> "is less than" <u>exp</u>
>=		<u>exp</u> "is greater to or equal to" <u>exp</u>
<=		<u>exp</u> "is less than or equal to" <u>exp</u>
AND		<u>exp</u> AND <u>exp</u> are both true
OR		either <u>exp</u> OR <u>exp</u> are true
NOT		NOT <u>exp</u> is true when <u>exp</u> is false, and false when <u>exp</u> is true.

MATRICES (ARRAYS) AND STRINGS

BASIC75 allows the use of one and two dimensional arrays, and character strings up to 256 characters in length. The statements and functions which take strings as arguments are described with the other statements and functions. Those which apply to arrays are listed here.

NOTE: Matrix variables must be a single letter from A to Z.

<u>NAME</u>	<u>SAMPLE STATEMENT</u>	<u>PURPOSE</u>
DIM	10 DIM A (10, 20)	Allocates space for a matrix of the specified dimensions.
MAT IDN	15 MAT X=IDN (m,m)	Establishes an identity matrix (with all ones down the diagonal). A new working size (m,m) may be specified.
MAT ZER	20 MAT B=ZER	Sets all elements of the specified matrix equal to 0.
	25 MAT D=ZER (m,n)	A new working size (m, n) may be specified after ZER.
MAT CON	30 MAT C=CON	Sets all elements of the specified matrix equal to 1.
	35 MAT E=CON (m,n)	A new working size (m,n) may be specified after CON.
MAT INPUT	40 MAT INPUT A(4,7)	Allows input of a matrix from the terminal; a new working size may be specified.
	45 INPUT A(5,5)	Allows input from the terminal of a specified matrix element.
MAT PRINT	50 MAT PRINT A	Prints the specified matrix on the terminal.
	55 PRINT A(X,c)	Prints the specified element of a matrix on the terminal; element specifications X and Y can be any expression.
	60 PRINT #2; A(1,5)	Prints matrix element on the specified file number.
	65 MAT PRINT #2,3;A	Prints matrix on a specified file and record.
MAT READ	80 MAT READ A	Reads matrix from DATA statements.
	85 MAT READ A(5,5)	Reads matrix of specified size from DATA statements.
	90 READ A(X,Y)	Reads the specified matrix element from a DATA statement.
MAT+	110 MAT C=A+B	Matrix addition; A, B and C must be the same size.
MAT-	120 MAT C=A-B	Matrix subtraction; A, B and C must be the same size.
MAT*	130 MAT C=A*B	Matrix multiplication; number of columns in A must equal number of rows in B.
MAT=	140 MAT A=B	Assigns values of B to A.
MAT TRN	150 MAT B=TRN (A)	Transposes an m by n matrix to an n by m matrix; matrix cannot be transposed into itself.
MAT INV	160 MAT C=INV (B)	Inverts a square matrix into a square matrix of the same size; matrix may be inverted into itself.
	170 MAT A=(3)*A	Matrix multiplies by a scalar number.

MATRICES (ARRAYS) AND STRINGS

BASIC75 allows the use of one and two dimensional arrays, and character strings up to 256 characters in length. The statements and functions which take strings as arguments are described with the other statements and functions. Those which apply to arrays are listed here.

NOTE: Matrix variables must be a single letter from A to Z.

<u>NAME</u>	<u>SAMPLE STATEMENT</u>	<u>PURPOSE</u>
DIM	10 DIM A (10, 20)	Allocates space for a matrix of the specified dimensions.
MAT IDN	15 MAT X=IDN (m,m)	Establishes an identity matrix (with all ones down the diagonal). A new working size (m,m) may be specified.
MAT ZER	20 MAT B=ZER	Sets all elements of the specified matrix equal to 0.
	25 MAT D=ZER (m,n)	A new working size (m, n) may be specified after ZER.
MAT CON	30 MAT C=CON	Sets all elements of the specified matrix equal to 1.
	35 MAT E=CON (m,n)	A new working size (m,n) may be specified after CON.
MAT INPUT	40 MAT INPUT A(4,7)	Allows input of a matrix from the terminal; a new working size may be specified.
	45 INPUT A(5,5)	Allows input from the terminal of a specified matrix element.
MAT PRINT	50 MAT PRINT A	Prints the specified matrix on the terminal.
	55 PRINT A(X,c)	Prints the specified element of a matrix on the terminal; element specifications X and Y can be any expression.
	60 PRINT #2; A(1,5)	Prints matrix element on the specified file number.
	65 MAT PRINT #2,3;A	Prints matrix on a specified file and record.
MAT READ	80 MAT READ A	Reads matrix from DATA statements.
	85 MAT READ A(5,5)	Reads matrix of specified size from DATA statements.
	90 READ A(X,Y)	Reads the specified matrix element from a DATA statement.
MAT+	110 MAT C=A+B	Matrix addition; A, B and C must be the same size.
MAT-	120 MAT C=A-B	Matrix subtraction; A, B and C must be the same size.
MAT*	130 MAT C=A*B	Matrix multiplication; number of columns in A must equal number of rows in B.
MAT=	140 MAT A=B	Assigns values of B to A.
MAT TRN	150 MAT B=TRN (A)	Transposes an m by n matrix to an n by m matrix; matrix cannot be transposed into itself.
MAT INV	160 MAT C=INV (B)	Inverts a square matrix into a square matrix of the same size; matrix may be inverted into itself.
	170 MAT A=(3)*A	Matrix multiplies by a scalar number.

A COMPLETE LIST OF COMMANDS FOR EVERGREEN'S TIMESHARING COMPUTER SYSTEM

Note: Commands are executed immediately; they do not require statement numbers. All commands can be abbreviated to three characters, as shown in the examples. The digit zero is slashed (Ø) so it is not confused with the letter "O".

COMMANDS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
APPEND	APP-PROG1 APP-\$PROG APP-PROG.E1Ø1	Append named program to current program, optionally retrieving named program from another user's catalog.
BYE	BYE	Logs user off the system.
CATALOG	CAT CAT-NAME CAT-.B4Ø4 CAT-NAME.B8Ø6	Lists the names and lengths of user programs and files, or those accessible from another ID, optionally beginning with the named entry.
CSAVE	CSA	Save the current program in semi-compiled form in user's library.
DELETE	DEL-1ØØ DEL-1ØØ,2ØØ	Delete all statements after & including the number. Delete all statements between & including the numbers.
DIRECTORY	DIR	Lists various information about each entry in user's library.
ECHO	ECH-OFF ECH-ON	Allows half-duplex; entered after log-in. Restores full-duplex mode.
FILE	FIL-FILE1 FIL-AFILE,1ØØ,2ØØ	Stores user program as ASCII strings in named file. Standard LIST parameters may be used.
GET	GET-PROG GET-\$PROG GET-PROG.C812	Retrieves named program and makes it the current program, optionally retrieving from another ID.
GROUP	GRO GRO-NAME	Lists the names and lengths of group programs and files, optionally starting with named entry.
HELLO	HEL-DØØ7,B ^c G ^c	Log on. User needs ID code and password.

COMMANDS

NAME	EXAMPLE	PURPOSE
RATE	RAT-120,3 RAT-,2 RAT-30	Alters the <u>port</u> speed and optionally subtype. <u>Terminal</u> speed must be changed by user. If carriage return is not received within two minutes, speed defaults back to original. Break in this time period will default speed immediately.
REFERENCE	REF-A\$ REF-B4 REF-PRINT REF-600 REF-A() REF-FNA	Prints a cross-reference of the specified variable (A\$,B4), statement type (PRINT), line number (600), array (A()), or user defined function (FNA).
RENUMBER	REN REN-50 REN-60, <u>n</u> REN- <u>m</u> , <u>n</u> , <u>p</u> , <u>q</u>	Renumbers program from 10 in intervals of 10. Renumbers program from specified statement number in intervals of 10. Renumbers program from specified statement number in intervals of <u>n</u> . Renumbers program starting at new statement number <u>m</u> , at intervals of <u>n</u> with renumbering starting at statement <u>p</u> and ending at statement <u>q</u> . <u>n</u> , <u>m</u> , <u>p</u> and <u>q</u> represent statement numbers
REPORT	REP	Prints a summary of system resources allocated and used by user.
RUN	RUN RUN-50	Starts current program executing, optionally starting at a specified line.
SAVE	SAV	Saves current program in user library.
SCRATCH	SCR	Deletes current program.
SYNTAX	SYN-FILE1	Reads ASCII strings off specified file and interprets them as BASIC code, adding them to current program.
TAPE	TAP	Informs system that following input is from paper tape.
TIME	TIM	Prints ID code and port number of current user, time of day, elapsed session wall time, and elapsed session CPU seconds.
XPUNCH	XPU	Produces same results as punch but adds an X-off character at the end of each line.

PROTECT LEVEL CHART

All users may assign any entry in their library one of four protect levels with the PROtect command. These levels are:

CR Creator status (default)	RP Release Protected
RL Release Locked	RF Release Free

This chart tells whether entries in another user's catalog are accessible by the various commands and statements. For example, by saying GET-TEST.E215, we can GET the program TEST from user E215 only if the program has been RP'ed or RF'ed by user E215.

IF STATUS IS:

Command or Statement	CR	RL	RP	RF
GET	NO	NO	YES	YES
INVoke, SPAWN, JOB	NO	YES	YES	YES
Modifications to a program	NO	NO	YES	YES
LIST, PUNCh, XPUnCh, SAVE, CSAve, FILE, APPend, REFerence	NO	NO	NO	YES
CHAIN	NO	YES, BUT NOT TO A LINE NUMBER	YES	YES

FILES

ASSIGN	NO	YES*	YES	YES
Reading from a file	NO	YES*	YES	YES
Writing on a file	NO	YES*	NO	YES
LOCK, UNLOCK	NO	YES*	NO	YES+

*If the program is release locked and from the same ID
+If user has write access

NOTE: A CATalog listing from another ID (eg. CAT-.E189) displays only programs and files that are RL, RP, or RF. No CR status entries are displayed to foreign users.

FUNCTIONS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
ABS	100 PRINT ABS(X)	Absolute value of X.
ATN	250 Y = ATN(X)	Arctangent of X radians.
CHR\$	100 A\$=CHR\$(65) 110 PRINT CHR\$(38)	Returns the ASCII character specified by the argument. (See ASCII Chart).
COS	210 Y = COS(X)	Cosine of X in radians.
ERR	120 PERFORM ERR(0) 121 IF ERR(-4) THEN DO	ERR(1) Enables printing of warning messages. ERR(0) Disables printing of warning messages. In the following test, value will be 1 if the specified condition occurred, otherwise 0. ERR(-1) Overflow ERR(-2) Underflow ERR(-3) Divide by zero ERR(-4) Undefined value ERR(-5) Log of zero ERR(-6) Zero to negative power ERR(-7) Exponential overflow
EXP	300 PRINT EXP(X)	The constant e (2.71828) raised to the power X.
FN	310 Y = FNA(X)	Function A (as defined in a DEF statement) is determined for value X.
INT	320 PRINT INT(X)	Gives the largest integer $\leq$ the expression X.
LEN	330 L = LEN(A\$)	Gives the current length of a string (A\$) in characters.
LIN	332 PRINT LIN(X); A	Prints X linefeeds. Can be used only in a PRINT statement.
LOG	340 A = LOG(X)	Gives the natural logarithm of an expression.
NUM	130 PRINT NUM(A\$) 140 X+NUM(A\$(5))	Returns the decimal equivalent of the first character in the string or substring.
SYS	205 A=SYS(X)	X=18 Returns the value in the switch register. X=17 Forces a swap-out. Returns the number of tenths of seconds to return to execution. X=16 Return the select-break character code X=15 Enable select-break. X=14 Disable select-break X=13 Log the user off X=12 ECHO On X=11 ECHO Off X=10 Disable kill-all-output X=9 Enable kill-all-output X=8 Enable control-C-exit X=7 Disable control-C-exit

FUNCTIONS

<u>NAME</u>	<u>EXAMPLE</u>	<u>PURPOSE</u>
		X=6 Amount of core space remaining X=5 No. of users logged on account X=4 Child interrupt INT(SYS(4)/256) = CHILD PORT# SYS(4) MOD 256 = Return Code X=3 Message Off X=2 Message On X=1 User's ID Character (A=1,B=2,...,Z=26) X=0 User's ID No. (1024*CHARACTER+NUMBER)
	210 C = SYS(-X)	SYS of a negative argument returns the contents of the core memory location addressed by X.
TAB	220 PRINT TAB(X);A\$;	Tabs to the specified print position(X). Can be used only in a PRINT statement.
TAN	225 Y = TAN(X)	Tangent of X in radians.
TIM	210 A=TIM(X)	X=3 Year of century (0-99) X=2 Day of year (1-366) X=1 Hour of day (0-23) X=0 Minute of hour (0-59) X=-1 Tenths of seconds (0-599) X=-2 Minute of hour (0-59) X=-3 Hour of day (0-23) X=-4 Day of month (0-31) X=-5 Month of year (1-12) X=-6 Year of century (0-99) X=-7 CPU seconds this session X=-8 Elapsed session time
TYP	510 Y = TYP(X)	If X < 0, gives the type of data in a file as: 0=Non-existent file, 1=numeric, 2=string, 3=end-of-file, 4=end-of-record. If X=0, gives the type of data in a DATA statement as: 1=numeric, 2=string, 3=out-of-data. If X > 0, gives the type of data in a file as: 1=number, 2=string, 3=end-of-file. Skips over end-of-record markers.
POS	150 X=POS(A\$,B\$)	Returns the first position in B\$ that A\$ occurs. Returns 0 if A\$ is not a substring of B\$.
POP	160 X=POP(1)	POP(+n) returns the nth element on the stack relative to the Q register. POP(0) returns the top of the stack and decrements the stack pointer. "stack" refers to the user defined data stack, not the GOSUB stack. See STACK.
PUSH	170 A=PUSH(X)	Pushes the value X onto the user defined stack.
REC	180 A=REC(X)	X > 0 Returns the current record of file #X. X < 0 Returns the number of records in file #X.

FUNCTIONS

RND	322 J=RND(X)	Generates a random number between 0 and 1. The argument X can be zero, positive, or negative. For details, see programming manual.
SGN	330 I=SGN(X)	Gives 1 if $X > 0$; 0 if $X = 0$; and -1 if $X < 0$.
SIN	335 S=SIN(X)	Sine of X in radians.
SPA	337 PRINT A;SPA(X);B	Prints X blank spaces. Can be used only in a PRINT statement
SQR	340 S2=SQR(X)	Square root of X.
STK	190 A=STK(X)	X= 7 Maximum current data stack size X= 6 Returns value of the Q register X= 5 Pointer to 3rd element from TOS X= 4 Pointer to 2nd element from TOS X= 3 Pointer to 1st element from TOS X= 2 Pointer to top of stack (TOS) X= 1 Returns top of GOSUB stack X= 0 Current line number X=-1 Pops GOSUB stack, returns number of remaining elements. X=-2 Maximum GOSUB stack size X=-3 Current GOSUB stack pointer X=-4 Increments top of GOSUB stack by one

HISTORICAL NOTES

The TXB.2 operating system and the BASIC75 programming language were designed, implemented, and documented entirely by undergraduate students at The Evergreen State College (TESC), Olympia, Washington. The project was supported by TESC Computer Services and carried out under the supervision of Mr. John D. Munro.

The students responsible for designing, implementing, and documenting TXB.2 and BASIC75 include Chas Douglass, James S. Bennett, William Clocksin, Carol Brown, and Ben Norton. Apologies to those omitted.

The system runs on a Hewlett-Packard 2000C'F at TESC.

This documentation is copywrited © 1976 by William Clocksin and Computer Services, The Evergreen State College.

Photocopying permitted. Not for resale in any form.